

sort()

ความสามารถ

- เป็น function ที่ทำหน้าที่เรียงลำดับข้อมูลใน array
- เรียงแบบ ascending (น้อยไปมาก)
- แปลงข้อมูลให้เป็น string ก่อนเรียง โดยเรียงตามลำดับตัวอักษรแบบ UTF-16

*ถ้าต้องการเรียงข้อมูลแบบ descending (มากไปน้อย) สามารถใช้ function reverse() แทนได้

Syntax

- **sort()**
 - ไม่ต้องมี parameter ก็สามารถใช้ array.sort() ได้เลย และจะคืนผลลัพธ์ออกมาเป็น array ที่เรียงข้อมูลแบบน้อยไปมาก ตามตัวอักษรให้กลับมา
 - ตัวอย่าง
 - Sort ตัวอักษร

```
let names = ["Mary", "Alice", "Peter", "Bob"];
names.sort();
console.log(names);
```

Output : ["Alice", "Bob", "Mary", "Peter"]
 - Sort ตัวเลข

```
let numbers = [5, 1, 3000, 92];
numbers.sort();
console.log(numbers);
```

Output : [1, 3000, 5, 92] //เรียงแบบ string ไม่ใช่ตัวเลข
//3000 น้อยกว่า 5 และ 92 เพราะ '3' น้อยกว่า '5' และ '9'
- **sort(compareFn)**
 - รับ parameter หนึ่งตัวเป็น function
 - ใช้ในการเรียงลำดับข้อมูลแบบตัวเลขแทน โดยการใช้ compare function
 - Compare function แบบ ascending order (น้อยไปมาก) มี syntax ดังนี้
function(a,b) {return a - b}
 - ข้อมูลใน array จะถูกส่งไปให้ compare function และ return ค่าไปให้ sort เพื่อเรียงข้อมูล

- ค่าลบ ถ้า $a < b$: a จะถูกเรียงก่อน b
- ค่าบวก ถ้า $b > a$: a จะถูกเรียงหลัง b
- ค่า 0 ถ้า $a = b$: ไม่มีการสลับที่ข้อมูล

○ ตัวอย่าง

■ Sort ตัวเลข

```
function compareNumber(a,b){
    return a - b;
}
let numbers = [5, 1, 3000, 92];
numbers.sort(compareNumber);
console.log(numbers);
Output : [1, 5, 92, 3000]
```

■ Sort ตัวอักษร

```
function compareName(a,b){
    if(a < b) {return -1;}
    if(b > a) {return 1;}
    return 0;
}
let names = ["Mary", "Alice", "Peter", "Bob"];
names.sort(compareName);
console.log(names);
Output : ["Alice", "Bob", "Mary", "Peter"]
```

■ Sort object

```
function compareName(a,b) {
    if (a.name < b.name){
        return -1;
    }
    if (a.name > b.name){
        return 1;
    }
    return 0;
}
let person = [
```

```

    {name : "Mary", lastname : "Janes"},
    {name : "Alice", lastname : "Wonderland"},
    {name : "Bob", lastname : "Bubbles"},
    {name : "Peter", lastname : "Parker"}
  ]
  person.sort(compareName);
  console.log(person);

```

Output : [{name:"Alice", lastname : "Wonderland"}, {name : "Bob", lastname : "Bubbles"}, {name : "Mary", lastname : "Janes"}, {name : "Peter", lastname : "Parker"}]

■ Sort array

```

function compareArray(a,b){
  return a[1]-b[1];
}
let arr1 = [[1,2],[3,2],[5,10],[4,3]];
arr1.sort(compareArray);
console.log(arr1);
Output : [ [ 1, 2 ], [ 3, 2 ], [ 4, 3 ], [ 5, 10 ] ]

```

● **sort(function compareFn(firstE1, secondE1)**

- ทำงานเหมือนกับ sort(compareFn) แต่เอา function มาเขียนเป็น parameter เลย
- ตัวอย่าง

■ Sort ตัวเลข

```

let numbers = [5, 1, 3000, 92];
numbers.sort(function compareNumber(a,b){
  return a - b;
});
console.log(numbers);
Output : [1, 5, 92, 3000]

```

● **sort((firstE1, secondE1) => {...})**

- ทำงานเหมือนกับ `sort(function compareFn(firstE1, secondE1)` แต่ใช้ arrow function ทำให้เขียนได้สั้นลง
- ตัวอย่าง

- Sort ตัวเลข

```
let numbers = [5, 1, 3000, 92];  
numbers.sort((a,b) => {a - b});  
console.log(numbers);  
Output : [1, 5, 92, 3000]
```